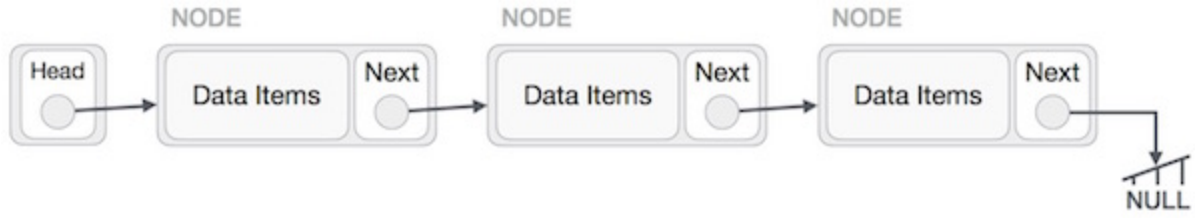
Date=04/08/2020
Lecture By=Shubham Joshi
Subject ⇒ Linked list

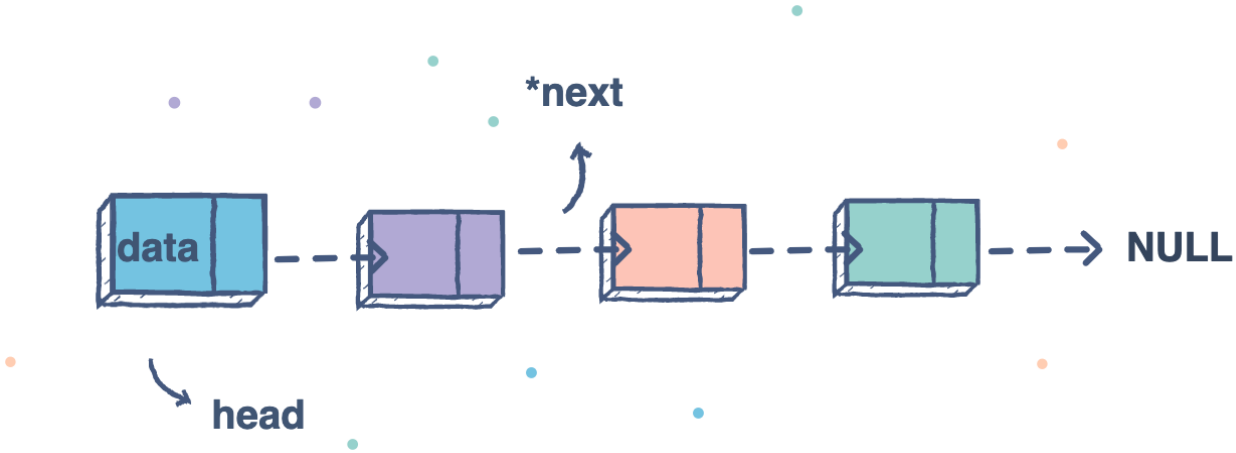
IN PREVIOUS LECTURE **(QUICK RECAP)** **Date-04/08/2020**	**In Today's Lecture (Overview)**
Principle Of Oops 1.Polymorphism 2.Encapsulation 3.Inheritance Super Function In Python Oops 4.Abstraction	What is linked list in python Important things What Is Node In Python MCQs Questions for self practice

What is linked list in python

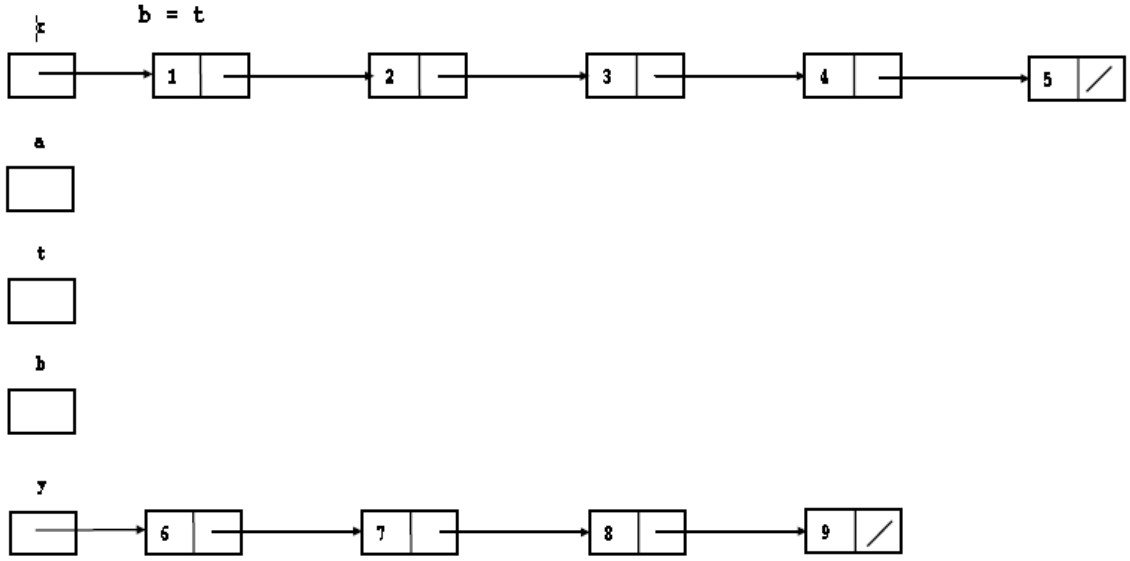
-A linked list is a sequence of data elements, which are connected together via links
=Each data element contains a connection to another data element in form of a pointer.
=Python does not have linked lists in its standard library. We implement the concept of linked lists using the concept of nodes
Complexity of linked list is O(n)

Linked list Representation






```
def mystery(a,b):
    while b != None:
        t = a.next.next
        a.next.next = b
        a = b
        b = t
```



Creation of linked list(Example)

```
class Node:
    def __init__(self, dataval=None):
        self.dataval = dataval
        self.nextval = None

class SLinkedList:
    def __init__(self):
        self.headval = None
```

```
  def listprint(self):
      printval = self.headval
      while printval is not None:
          print (printval.dataval)
          printval = printval.nextval

list = SLinkedList()
list.headval = Node("Mon")
e2 = Node("Tue")
e3 = Node("Wed")

# Link first Node to second node
list.headval.nextval = e2

# Link second Node to third node
e2.nextval = e3

List.listprint
```

Output

```
E:\Study\Codes\if>
Mon
Tue
Wed
```

Example which was explained in the lecture

```
class Node:
    def __init__(self, val):
        self.val = val
        self.next = None

def printLinkedList(head):
    cur = head
    while cur != None:
```

```
        print(cur.val)
        cur = cur.next

def addElementToEndOfTheLL(head, x):
    if head is None:
        return Node(x)

    cur = head
    while cur.next != None:
        cur = cur.next
    cur.next = Node(x)
    return head

def addElementInBetweenLL(head, after_which, x):
    cur = head

    while cur.val != after_which:
        cur = cur.next

    new_node = Node(x)
    new_node.next = cur.next
    cur.next = new_node

    return head

if __name__ == '__main__':
    head = None # 5->15->20->25->30
    head = addElementToEndOfTheLL(head, 5)
    head = addElementToEndOfTheLL(head, 15)
    head = addElementToEndOfTheLL(head, 20)
    head = addElementToEndOfTheLL(head, 25)
    head = addElementToEndOfTheLL(head, 30)
    #printLinkedList(head)
    addElementInBetweenLL(head, 20, 100)
    printLinkedList(head)
```

Output

```
E:\Study\Codes\if>C:/python/p
5
15
20
100
25
30
```

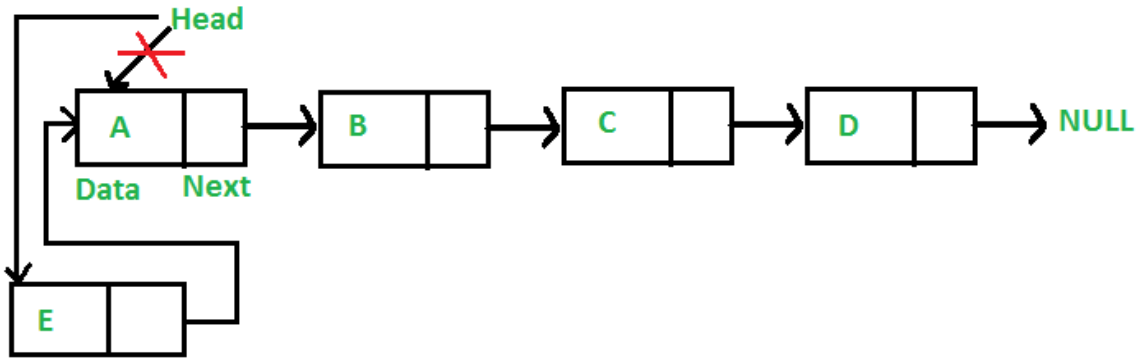
"Click Here" To Know More About it

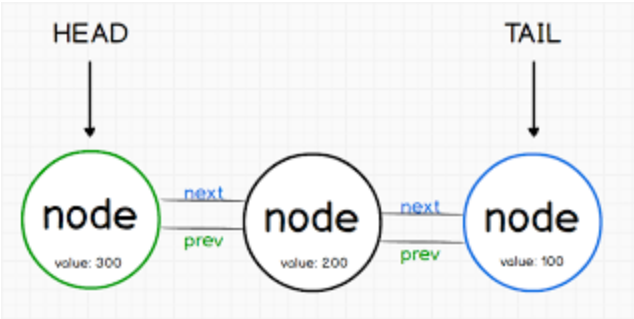
Important things

=Linked List Works Like A Chain
= First Element Of list is Called Head And Rest Are Called Tails

What Is Node In Python

A **linked list** is a linear data structure where each element is a separate object
Each Element of list is called node





MCQs

1.What is the time complexity to iterate a linked list

a.O(nlogn)

b.O(n2)

c.O(n)

2.what is the complexity of inserting an element at the start of the linked list ?

a. O(n)

b.O(nLogn)

c.O(1)

d.O(n2)

3.What is the complexity to add an element at the middle of an array ?

a.O(n2)

b.O(1)

c.O(n)

4.what is true about linked list ?

a.the memory addresses are random in it

b.the memory addresses are continuous in it

Questions for self practice

Q1. https://leetcode.com/problems/middle-of-the-linked-list/
Q2.https://leetcode.com/problems/design-linked-list/